A Topological Perspective on Diagnosis

Andreas Bauer, *ANU, Australia* Sophie Pinchinat, *IRISA, France*

***Abstract*— We propose a topological perspective on the diagnosis problem for discrete-event systems. In an infinitary framework, we argue that the construction of a centralized diagnoser is conditioned by two fundamental properties: *saturation* and *openness*. We show that these properties are decidable for ω-regular languages. Usually, openness is guaranteed implicitly in practical settings. In contrast to this, we prove that the saturation problem is PSPACE-complete, which is relevant for the overall complexity of diagnosis.**

I. Introduction

The diagnosis of (discrete-event) systems, originally formalized in [20], consists in establishing a verdict on the status of the actual computation of the system regarding a given property, based on external observable events of this computation.

The system under consideration is assumed to be finite state and the property under scrutiny is given by a regular language over the behaviors of the system. Traditionally, this regular property is described by a finite-state automaton [7], but logical specifications have also been considered [9].

The central difficulty in diagnosis problems is the imperfect information about the computations. Given a stream of observations, there are in general several behaviors of the systems that are consistent with this information. To know with certainty that the actual computation has the desired property, one needs to make sure that the entire set of consistent behaviors have this property. This is precisely what *diagnosability* is about. Several procedures to decide diagnosability have been studied. The most efficient one is based on the *twin plant construction* [8], which runs in quadratic time when the property is described by a deterministic finite state automaton. Diagnosability is proved to be a necessary and sufficient condition for the existence of a bound in the number of observations that are needed to establish that the property holds. On the other hand, when the property is specified in a logical formalism, diagnosability is not sufficient anymore to guarantee that a verdict on the computation can be delivered in a finite amount of time. This can be achieved only when the system satisfies an additional property called *prediagnosability* [9].

In this paper, we characterize diagnosability and prediagnosability in terms of classical mathematical concepts: *saturation* and *openness* in the topological space of infinite words. We investigate the associated decision problems and analyze their complexity.

This research was supported by the Marie Curie Scientific Project MASLOG 021669 (FP6-2004-Mobility-6) and Univ. de Rennes 1.

Saturation involves a language and an equivalence relation: The language is saturated if it does not distinguish between equivalent elements. We give a PSPACE decision procedure for the saturation problem when the language is ω-regular and the equivalence relation is ω-rational. Our method is inspired from [14]. We specialize this general problem to the case of the observational equivalence between words induced by the imperfect information setting and show that this instance of the saturation problem, which corresponds precisely to diagnosability, is PSPACE-complete.

The paper is organized as follows: in Sec. II we present notations and vocabulary and in Sec. III we introduce topological concepts in the space of infinite words such as openness and the central notion of saturation. As a preliminary to our decision procedure for saturation, we dedicate the Sec. IV to rational relations between words and to 2-automata. In Sec. V, we present the decision procedures for saturation and openness, and analyze their complexity. We conclude the contribution with a comparison of our approach with existing work on centralized diagnosis.

II. Elementary notations and vocabulary

Given an alphabet $\Sigma = \{a, b, l, \ldots\}$, we denote by Σ^* (resp. Σ^ω) the set of finite (resp. infinite) words over Σ. We use $u, u', v, \ldots$ (resp. $w, w', w_1, \ldots$) as typical elements of Σ^* (resp. Σ^ω). For $w \in \Sigma^\omega$, we write w_k for the k-th prefix of w. A $*$-language (resp. ω-language) is any subset of Σ^* (resp. Σ^ω) – we will indifferently use "language" and "set" –. We use $B, B', \ldots$ (resp. $L, L', S, \ldots$) for typical $*$-languages (resp. ω-languages). For any $L \subseteq \Sigma^\omega$, let us denote by L^c the complement of L, that is $\Sigma^\omega \setminus L$. Given a set $B \subseteq \Sigma^*$, we denote by $B\Sigma^\omega$ the set of words of the form uw with $u \in B$ and $w \in \Sigma^\omega$.

We fix a distinguished subset Σ_o of Σ. Elements of Σ_o are called *observables* and elements of its complement Σ_{uo}, are *unobservables*. Typical elements of Σ_o^* (resp. Σ_o^ω) are $\tau, \tau', \tau_1, \ldots$ (resp. π, π', π_1). Any word over Σ_o is an *observation*. We denote by P the projection on words which transforms a word over Σ into a word over Σ_o by erasing every element of Σ_o. From the projection P, we derive an equivalence relation $\approx$ between words, called *the observational equivalence*, which identifies two words whenever their P-images coincide and they are both either finite or infinite. Given any equivalence $\sim$ between finite (resp. infinite) words, we denote by $[u]_\sim$ (resp. $[w]_\sim$) the equivalence class of u (resp. w).

III. Topologies on the Space of Infinite Words

The *Cantor topology* over the set Σ^ω of infinite words is defined as follows: the basic open sets are the sets of the form $B\Sigma^\omega$ where $B \subseteq \Sigma^*$ (see [15, Chapter 3]). A set is *closed* if its complement is open. It is *clopen* is it is both open and closed. Clopen sets are of the form $B\Sigma^\omega$ where B is a finite subset of Σ^*.

We refine the Cantor topology with respect to a fixed equivalence relation between words. This is done by enforcing the open sets to be "saturated" by the equivalence relation. In the rest of the section we fix an equivalence relation $\sim$ over Σ^ω.

Definition 1: The $\sim$*-saturation* of an ω-language L is the ω-language $(L)_\sim$ defined by $(L)_\sim := \bigcup_{w\in L}[w]_\sim$.
L is$\sim$*-saturated*, whenever $L = (L)_\sim$.

In Sec. V-A, we examine a procedure to decide whether a language is $\sim$-saturated.

Lemma 2: Complement, union and intersection preserve $\sim$-saturation.

We consider a new topology on Σ^ω where the open sets are $(B\Sigma^\omega)_\sim$ with $B \subseteq \Sigma^*$. They are the $\sim$-saturations of open sets in the Cantor topology.

As words corresponding to computations of a given (discrete-event) system are not in general arbitrary elements of Σ^ω, we consider the topology induced by some fixed ω-language $S \subseteq \Sigma^\omega$. The language S is called a *system*. In the topology induced by S, (Cantor) open sets are of the form $B\Sigma^\omega \cap S$ and $\sim$-open sets are of the form $(B\Sigma^\omega)_\sim \cap S$ (where $B \subseteq \Sigma^*$). A language $L \subseteq \Sigma^\omega$ is *open* (resp. $\sim$*-open*) *in* S whenever $L \cap S = B\Sigma^\omega \cap S$ (resp. $= (B\Sigma^\omega)_\sim \cap S$). Notice that an open in (resp. $\sim$-open) set in S is not necessarily open (resp. $\sim$-open) in the Cantor topology.

We say that L *is* $\sim$*-saturated in* S whenever

$$L \cap S = (L)_\sim \cap S$$

Note that $\sim$-saturation and $\sim$-saturation in Σ^ω coincide. Being $\sim$-saturated in S implies being $\sim$-saturated in S' for any $S' \subseteq S$, but that the converse does not hold in general. To see this, consider any situation with the following strict inclusions $S' \subset L \subset S = (L)_\sim$.

Lemma 3: If a set is $\sim$-open in S then it is $\sim$-saturated in S.

Proof: It is sufficient to show that for any $B \subseteq \Sigma^*$, $((B\Sigma^\omega)_\sim \cap S)_\sim \cap S = (B\Sigma^\omega)_\sim \cap S$. We only establish $((B\Sigma^\omega)_\sim \cap S)_\sim \cap S \subseteq (B\Sigma^\omega)_\sim \cap S$, as the reverse inclusion is immediate. Let $w \in ((B\Sigma^\omega)_\sim \cap S)_\sim \cap S$, then $w \in S$ and it remains to prove that $w \in (B\Sigma^\omega)_\sim$. Because by assumption $w \in ((B\Sigma^\omega)_\sim \cap S)_\sim$, there exists $w' \sim w$ with $w' \in (B\Sigma^\omega)_\sim \cap S$. In particular, $w' \in (B\Sigma^\omega)_\sim$. There must exist $w'' \in B\Sigma^\omega$ such that $w'' \sim w'$. Since $\sim$ is transitive, $w \sim w''$ which entails $w \in (B\Sigma^\omega)_\sim$. ∎

Lemma 4: An open set is $\sim$-saturated in S if and only if it is $\sim$-open in S.

Proof: Consider a language $B\Sigma^\omega$ with $B \subseteq \Sigma^*$. It is $\sim$-saturated in S if and only if $B\Sigma^\omega \cap S = (B\Sigma^\omega)_\sim \cap S$, which shows that it is $\sim$-open in S. For the reciprocal, apply Lem. 3. ∎

In the rest of this section, we focus on the case where $\sim$ is $\approx$, the observational equivalence. The following lemma shows that the equivalence $\approx$ is very particular:

Lemma 5: For any $B \subseteq \Sigma^*$

$$(B\Sigma^\omega)_\approx = (B)_\approx\Sigma^\omega \tag{1}$$

where $(B)_\approx = \bigcup_{u\in B}[u]_\approx$.

Lemma 6: A language is $\approx$-open in S if and only if it coincides on S with an open and $\approx$-saturated language.

Proof: Assume a language L is $\approx$-open in S. Since L satisfies $L \cap S = (B\Sigma^\omega)_\approx \cap S$ and by (1) this language coincides on S with an open set, namely $(B)_\approx\Sigma^\omega$.

Assume $L \cap S = L' \cap S$ where L' is open and $\approx$-saturated. We apply Lem. 4 to L' to rewrite $L' \cap S$ as $(B\Sigma^\omega)_\approx \cap S$ for some $B \subseteq \Sigma^*$, which shows that L is $\approx$-open in S.

∎

We now establish Prop. 7 and Cor. 8 as essential results for the theory of discrete-event systems diagnosis.

To every $\tau \in \Sigma_o^*$ we associate the set $\langle\tau\rangle := P^{-1}(\tau)\Sigma^\omega$. By definition, $\langle\tau\rangle$ is open, and it is $\approx$-saturated by construction; hence it is $\approx$-open by Lem. 4. Note that an infinite word $w \in \langle\tau\rangle$ whenever w is of the form $w = uw'$ for some u with $P(u) = \tau$. We will abuse notation by writing $\langle\pi\rangle$ to mean the set of infinite words w such that $P(w) = \pi$.

Proposition 7: Let $L \subseteq \Sigma^\omega$. L is $\approx$-open if and only if for every observation $\pi \in \Sigma_o^\omega$ with $L \cap \langle\pi\rangle \neq \emptyset$, there exists $k(\pi) \in \mathbb{N}$ such that $\langle\pi_{k(\pi)}\rangle \subseteq L$.

Proof: $\Leftarrow$) is immediate since the $\langle P(w)_{k(P(w))}\rangle$'s are $\approx$-open sets and $L = \bigcup_{w\in L}\langle P(w)_{k(P(w))}\rangle$.

$\Rightarrow$) By Lem. 6 (for $S = \Sigma^\omega$) L is open and $\approx$-saturated, and by the later, $L \cap \langle\pi\rangle \neq \emptyset$ implies

$$\langle\pi\rangle \subseteq L \tag{2}$$

For the readability of the remaining we simply write k for $k(\pi)$. Assume that for each $k \geq 0$ there exists $w'_k \in \langle\pi_k\rangle \setminus L$. Since L is open, it is of the form $B\Sigma^\omega$ with $B \subseteq \Sigma^*$. Since $w'_k \notin L$, the j-th prefix $(w'_k)_j$ of w'_k is not B, for every $j \in \mathbb{N}$. Because the alphabet Σ is finite, we apply Koenig's lemma to the set $\{(w'_k)_j \mid j, k \in \mathbb{N}\}$ and obtain an infinite sequence of elements $u_0 < u_1 < u_2...$ such that for every i, $P(u_i)$ is the i-th prefix of π. Since the sets $\{u_i\}\Sigma^\omega$ are clopen sets, their intersection is closed and therefore contains the limit of the u_i's, say w', which lies outside the set L. But $P(w') = \pi$, since each $P(u_i)$ is a prefix of π, hence $w' \in \langle\pi\rangle$, which contradicts (2).

∎

We assume given a system $S \subseteq \Sigma^\omega$. For $\tau \in \Sigma_o^*$, $\langle\tau\rangle_S$ denotes the set $\langle\tau\rangle \cap S$; note that $\langle\tau\rangle_S$ is $\approx$-open in S.

Corollary 8: Let $L \subseteq \Sigma^\omega$. L is $\approx$-open in S if and only if for every observation $\pi \in \Sigma_o^\omega$ with $L \cap \langle \pi \rangle_S \neq \emptyset$, there exists $k(\pi) \in \mathbb{N}$ such that $\langle \pi_{k(\pi)} \rangle_S \subseteq L$.

Proof: We simply write k for $k(\pi)$. Assume L is $\approx$-open in S. Then $L \cap S = (B\Sigma^\omega)_\approx \cap S$ for some $B \subseteq \Sigma^*$. Let $\pi \in \Sigma_o^\omega$ be such that $L \cap \langle \pi \rangle_S \neq \emptyset$. Since $L \cap S \subseteq (B\Sigma^\omega)_\approx \cap S$, we also have $(B\Sigma^\omega)_\approx \cap \langle \pi \rangle_S \neq \emptyset$. By applying Prop. 7 to the $\approx$-open $(B\Sigma^\omega)_\approx$, π has some prefix π_k with $\langle \pi_k \rangle \subseteq (B\Sigma^\omega)_\approx$. Hence $\langle \pi_k \rangle_S \subseteq L$.

Reciprocally, assume that for every $\pi \in \Sigma_o^\omega$, $L \cap \langle \pi \rangle_S \neq \emptyset$ implies there exists $k \in \mathbb{N}$ such that $\langle \pi_k \rangle_S \subseteq L$. We want to show that L is of the form $L = (B\Sigma^\omega)_\approx \cap S$ for some $B \subseteq \Sigma^*$. The candidate for B is $\bigcup_{\pi \in \Sigma_o^\omega} P^{-1}(\pi_k)$.

By Lem. 5, $B\Sigma^\omega$ is $\approx$-saturated, so that $B\Sigma^\omega = (B\Sigma^\omega)_\approx$. It remains to show that $L \cap S = B\Sigma^\omega \cap S$:

- $L \cap S \subseteq B\Sigma^\omega \cap S$: Let $w \in L \cap S$, and let $\pi = P(w)$. Since $w \in P^{-1}(\pi_k)\Sigma^\omega \cap S$ and $P^{-1}(\pi_k) \subseteq B$, $w \in B\Sigma^\omega \cap S$.
- $B\Sigma^\omega \cap S \subseteq L \cap S$: Let $w \in B\Sigma^\omega \cap S$, then $w \in P^{-1}(\pi_k)\Sigma^\omega = \langle \pi_k \rangle$ which by hypothesis is contained in L and we are done.

■

In Sec. V we investigate decidability of $\sim$-saturation, openness, and $\approx$-openness. Elementary notions on rational relations are required to present the solution.

IV. Rational Relations

The class of *rational* relations and its subclasses ranging from *recognizable* relations to e.g. *synchronized* relations are of particular interest since they possess acceptors whose e.g. emptiness can be decided. A detailed literature on the topic can be found in [2], [25], [5], [16], [4].

For the purpose of this work, we will focus on binary relations only, that is pairs of words over the alphabet Σ, and we simply call them *relations*.

For finite words, a relation is a subset of the Cartesian product $\Sigma^* \times \Sigma^*$. The notions needed in the contribution are listed below.

A subset R of $\Sigma^* \times \Sigma^*$ is *rational* whenever it is an element of the set $\mathtt{Rat}_{\Sigma^* \times \Sigma^*}$, or simply $\mathtt{Rat}$, defined as the least subset of $2^{\Sigma^* \times \Sigma^*}$ such that: (1) every finite subset of $\Sigma^* \times \Sigma^*$ is in $\mathtt{Rat}$, (2) if $R, R' \in \mathtt{Rat}$, then $R \cup R', RR' \in \mathtt{Rat}$, and (3) if $R \in \mathtt{Rat}$, then $R^* := \bigcup_{i \geq 0} R^i \in \mathtt{Rat}$.

The following properties are decidable.

- Emptiness: for any rational ω-relation $\rho \subseteq \Sigma^\omega \times \Sigma^\omega$, one can construct non-deterministic Buchi automata which accept the first and the second projection of ρ. Now, ρ is empty if and only if either one of the two projections is empty. Buchi automata emptiness is decidable.
- Finteness: whether a rational ω-relation is finite amounts to checking that the two projections are finite ω-regular languages.

According to the general theory, *recognizable* relations over Σ are particular cases of recognizable subsets of a monoid (see [2, Chapter III]). In the case of the monoid $\Sigma^* \times \Sigma^*$ (for binary relations), we use the following intuitive characterization due to [3], known as the Mezei's Theorem: a *recognizable* relation $R \subseteq \Sigma^* \times \Sigma^*$ is a finite union of sets of the form $B_i \times B_i'$, where $B_i, B_i' \subseteq \Sigma^*$ are regular $*$-languages. As the monoid $\Sigma^* \times \Sigma^*$ is implicit in this work, we simply write $\mathtt{Rec}$ for the set of recognizable subsets of $\Sigma^* \times \Sigma^*$.

It is well established that $\mathtt{Rec} \subseteq \mathtt{Rat}$. However, because the monoid $\Sigma^* \times \Sigma^*$ is not free, the reciprocal does not hold.

Closure properties of the classes $\mathtt{Rat}$ and $\mathtt{Rec}$ are as follows: $\mathtt{Rec}$ is closed under union, intersection and complement, whereas $\mathtt{Rat}$ is closed under union (by definition) but not under intersection in general, therefore neither under complementation. One way to achieve good closure properties while maintaining expressiveness is by mixing the two classes. The following lemma from [2, Proposition 2.6, Chapter III] is at the basis of our analysis.

Lemma 9: If $R \in \mathtt{Rat}$ and $R' \in \mathtt{Rec}$, then $R \cap R' \in \mathtt{Rat}$.

Operationally, rational ω-relations are characterized by *2-automata* [17].

Definition 10: A *2-automaton* over the alphabet Σ is a structure $\Theta = (S, \Sigma, s_0, t, A, S_1, S_2)$ where (S, Σ, s_0, t, A) is a Buchi automaton with acceptance set $A \subseteq S$, and where $\{S_1, S_2\}$ is a partition of S into control states for the first and second input tape respectively.

A 2-automaton reads a pair of words (w_1, w_2) each placed on a distinct input tape; the partition $\{S_1, S_2\}$ tells which tape is to be read.

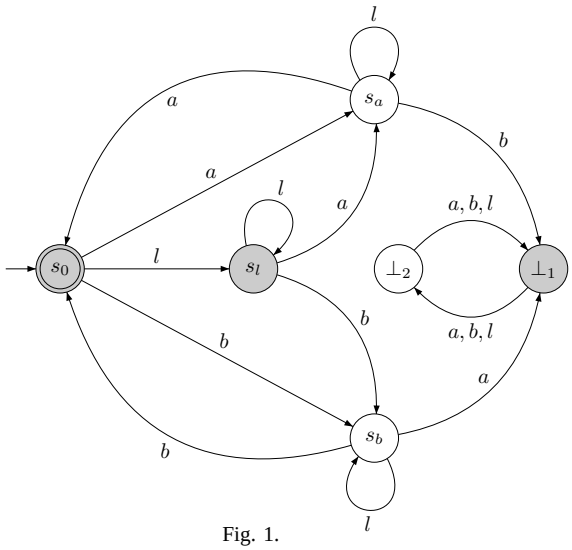

Fig. 1.

We illustrate the behavior of the 2-automaton in Fig. 1 characterizing the observational equivalence induced by $\Sigma_o = \{a, b\} \subseteq \Sigma = \{a, b, l\}$ for a pair of input words (w_1, w_2) of the form $w_1 = alblaaw_1'$ and $w_2 = llaballaw_2'$ (states filled in grey are in S_1). As the initial state s_0

belongs to S_1, the automaton first reads a from w_1 on tape 1, then it moves to state $s_a \in S_2$ to remember that the last observable read on tape 1 was a. It then reads l of w_2 on tape 2, which causes a transition back to s_a. Eventually, the first a of w_2 is read, and since this matches the expected observable, the automaton moves back to s_0. After a few more transitions, the automaton is in state s_0 and the tapes contain w_1' and w_2' respectively. Assume that $w_1' \not\approx w_2'$, henceforth $w_1 \not\approx w_2$, because say w_1' starts with an a whereas w_2' starts with a b. In this case, the run of the automaton gets trapped inside the non accepting maximal strongly connected component $\{\perp_1, \perp_2\}$. Notice that the state $s_l \in S_1$ is meant to "absorb" unobservables on tape 1; as it is not accepting, only words with a finite number of consecutive unobservables are recognized.

We remark that any observational equivalence can be characterized by a 2-automaton of size in $O(|\Sigma_o| + 4)$.

Notice that for a 2-automaton accepting an observational equivalence relation, the distance between the two heads cannot be bounded in general, because of unobservables. In other words, observational equivalences are not *synchronized* rational relations [4]. Nevertheless, their complement is rational.

V. Decision Problems

We address the problems of saturation and openness of ω-regular languages described by non-deterministic Buchi automata.

A. Deciding saturation

In the following, let $L(\Theta) \subseteq \Sigma^\omega \times \Sigma^\omega$ denote the language accepted by a 2-automaton Θ.

Theorem 11: Fix a non-deterministic Buchi automaton $\mathcal{A} = (Q, \Sigma, \delta, q_0, F)$ and a 2-automaton Θ whose language $L(\Theta)$ is an equivalence relation. The problem of whether $L(\mathcal{A})$ is $L(\Theta)$-saturated is PSPACE-complete.

Proof: Simply write L for $L(\mathcal{A})$ and $\sim$ for $L(\Theta)$.

i) PSPACE membership: let $R := \sim \cap (L \times L^c)$. Clearly, $R = \emptyset$ if and only if L is $\sim$-saturated, as both conditions mean that there are no two words $w_1 \sim w_2$ with $w_1 \in L$ and $w_2 \notin L$. By Mezei's Theorem, the relation $L \times L^c$ is recognizable since L and L^c are ω-regular. Since $\sim$ is rational, so is R (see a variant of Lem. 9 for ω-relations in [5]) and a 2-automaton can be effectively constructed for R.

We present an algorithm to check emptiness of R.

Let $\mathcal{B}$ be a non-deterministic Buchi automaton which accepts L^c using $O(2^{|\mathcal{A}| \log |\mathcal{A}|})$ states. This can be constructed following [10] (see also [6, Chapter 4]).

Let Θ' be the 2-automaton which behaves like Θ but whose input is componentwise constrained by $\mathcal{A}$ and $\mathcal{B}$ respectively. The 2-automaton Θ' has $O((|\Sigma_o| + 4) \cdot |\mathcal{A}| \cdot 2^{|\mathcal{A}| \log |\mathcal{A}|})$ states encoded in space $O(\log(|\Sigma_o| + 4)) + \log |\mathcal{A}| + |A| \log |\mathcal{A}|)$.

The following non-deterministic algorithm **A1** finds an accepting run of Θ', namely a sequence of states $r_0 r_1 \dots r_i \dots r_n$ where r_0 is an initial state, r_i is an accepting state, and $r_i = r_n$.

Algorithm A1

1) Let r be the initial state of Θ'
2) Choose a state r'
3) If r' is a successor of r, let $r = r'$
   else halt (without accepting)
4) If r is accepting, goto 5 or 2, else goto 2
5) Let $r_A = r$ // guess it is r_i
6) Choose a state r'
7) If r' is a successor of r, let $r = r'$
   else halt (without accepting)
8) If $r = r_A$, accept, else goto 6

This algorithm can be implemented by a non-deterministic polyspace Turing machine, which concludes since NPSPACE = PSPACE by Savitch Theorem [22].

ii) PSPACE hardness: let us denote by $\equiv$ the trivial relation $\Sigma^\omega \times \Sigma^\omega$. Given a Buchi automaton $\mathcal{C}$, we reduce the universality problem for $\mathcal{C}$ (whether $L(\mathcal{C}) = \Sigma^\omega$), known to be PSPACE-complete [23], to the $\equiv$-saturation of $L(\mathcal{C})$.

If $L(\mathcal{C}) = \emptyset$, which can be checked linearly in the size of $\mathcal{C}$, then return "no". Otherwise, let $w_1 \in L(\mathcal{C})$. $\mathcal{C}$ is not universal if and only there exists $w_2 \notin L(\mathcal{C})$. Since $w_1 \equiv w_2$ (because $\equiv$ is trivial) this is equivalent to saying that $L(\mathcal{C})$ is not $\equiv$-saturated. ■

When an ω-regular system S is considered, Algorithm **A1** can be easily adapted to decide $\sim$-saturation of L in S by checking emptiness of R intersected with the recognizable relation $S \times S$.

Corollary 12: The problem of checking $\approx$-saturation in S is PSPACE-complete.

Proof: For membership, use the fact that the set of states of the 2-automaton Θ' can be encoded in space $O(log(|\Sigma_o| + 4)) + log|\mathcal{A}| + |A|log|\mathcal{A}|)$. For hardness, take $S = \Sigma^\omega$ and $\Sigma_o = \emptyset$ and consider the proof for hardness in Th. 11, as in this case $\approx$ and $\equiv$ match. ■

B. Deciding openness

We recall that openness implies openness in S, for any $S \subseteq \Sigma^\omega$, but that the converse does not hold in general. However, it is easy to show that if S is a closed set, L is open in S if and only if $L \cup S^c$ is open.

Regarding openness, [12] proposes a polynomial procedure to decide whether the language of a deterministic Muller automaton (MA), see for example [6], is an open set. Deterministic MA are sufficiently expressive enough to cover the class of ω-regular languages, and in contrast to Buchi automata, they can always be assumed deterministic. Basically, a (deterministic) MA $\mathcal{M}$ is given by a finite state automaton (Q, Σ, q_0, δ) and a distinguished set $\mathcal{F} \subseteq 2^Q$ of *accepting sets*. An infinite word is accepted by $\mathcal{M}$ if along

the (unique) run of $\mathcal{M}$ for this input word, the set of states that are visited infinitely often matches some element of $\mathcal{F}$. In an MA $\mathcal{M}$, the set of accepting sets $\mathcal{F}$ consists of non trivial strongly connected components.

For non-deterministic Buchi automata (NBA), the openness problem is more complex. It is obviously decidable by translating the NBA into a deterministic MA [13], but with an exponential blow-up. Alternatively, by complementing the NBA, one can use the procedure of [1] which determines whether a language of an NBA $\mathcal{A}$ is closed [1]: the method consists in verifying that $\mathcal{A}$ and its *closure* (obtained by making all co-reachable states accepting) denote the same language. According to [24] this verification problem is PSPACE-complete in general, and it is linear time if $\mathcal{A}$ is deterministic [11]. However, the preliminary complementation of the NBA has an exponential cost.

To our knowledge, no lower bound for the openness problem of NBA has been established.

VI. Application to Diagnosis

In this section we consider the problem of diagnosing arbitrary ω-regular languages. The infinitary setting is worth considering as it brings insight into existing work on diagnosis. The reader familiar with classic diagnosis may be puzzled by the definition according to Eq. (3) as it does not match the standard definition of the diagnosis function, as in [7]. Differences and similarities between the infinitary and the finitary settings are discussed in Sec. VI-B.

A. Diagnosis of ω-languages

Assume given two ω-languages S and L. We want to diagnose that a word of S belongs to L on the basis of the observations of its prefixes; traditionally, elements of L are called *faulty sequences*. The diagnoser is fed incrementally with the sequence of finite observations of an infinite word of S. Let $\tau \in \Sigma_o^*$ be an observation. Assume a situation where for every possible infinite continuation π' of τ, any concrete scenario $w \in S$ consistent with $\tau\pi'$, that is $w \in \langle \tau\pi' \rangle_S$, belongs to L. In such a situation, membership of the actual computation of the system in L can be safely declared/predicted, i.e. we set a positive verdict ($\top$). Dually, if every scenario consistent with any possible infinite continuation of the observation does not belong to L, we set a negative verdict ($\bot$). The remaining cases are "confused" situations since scenarios consistent with the observation spread inside and outside L; in this case, the diagnoser returns ?.

According to the possible situations described above, we define the *diagnosis function* $\mathrm{Diag}_L : \Sigma_o^* \to \{\top, \bot, ?\}$ by

$$\mathrm{Diag}_L(\tau) := \begin{cases} \top & \text{if } \langle \tau \rangle_S \subseteq L \\ \bot & \text{if } \langle \tau \rangle_S \cap L = \emptyset \\ ? & \text{otherwise.} \end{cases} \quad (3)$$

We shall omit the subscript L when clear from the context.

For a diagnoser to be useful, one expects a positive verdict to be eventually delivered if the actual computation of the system is a faulty sequence.

Theorem 13: L is $\approx$-open in S if and only if for every $w \in L$, there exists $k(w) \in \mathbb{N}$ s.t. $\mathrm{Diag}(P(w)_{k(w)}) = \top$.

Proof: By definition of the diagnosis function, $\mathrm{Diag}(P(w)_{k(w)}) = \top$ is equivalent to $\langle P(w)_{k(w)} \rangle \subseteq L$. We can apply Cor. 8 for $\pi = P(w)$ to conclude. ■

Note that $\approx$-openness in S guarantees only issuing of positive verdicts. Regarding non faulty sequences, nothing can be inferred in a finite amount of time in general, unless L^c is $\approx$-open in S: Th. 13 then applies so that a negative verdict eventually occurs when observing a non faulty sequence. To be more specific, if L is $\approx$-clopen in S, confused situations cannot last.

Theorem 14: L is $\approx$-clopen in S if and only if for all $w \in S$, there exists $k(w) \in \mathbb{N}$ such that $\mathrm{Diag}(P(w)_{k(w)}) \neq ?$.

Proof: $\Rightarrow$) Let $w \in \Sigma^\omega$. If $w \in L$, because L is $\approx$-open in S, we can apply Th. 13 to infer the existence of $k(w)$. Otherwise $w \in S \setminus L$. Since L is $\approx$-closed in S, L^c is $\approx$-open in S. We therefore can apply Th. 13 to L^c and obtain the existence of $k(w)$ such that $(P(w)_{k(w)}) \subseteq L^c$, that is $\mathrm{Diag}(P(w)_{k(w)}) = \bot$.
$\Leftarrow$) Consider the partition of S into L and $S \setminus L$. For every $w \in L$, $\mathrm{Diag}(P(w)_k) \neq ?$ is equivalent to $\mathrm{Diag}(P(w)_k) = \top$. By Th. 13, L is $\approx$-open in S. By a similar reasoning, $S \setminus L$ is $\approx$-open in S, hence L is $\approx$-closed in S. ■

B. Comparison with classic diagnosis

The classic diagnosis framework [21] deals with finite words. A discrete-event system $\mathcal{S}$ can be viewed as a finite state deterministic automaton over an alphabet (say Σ) with all states marked as final. As presented in a most general setting in [7], a *supervision pattern* Ω is a regular $*$-language which is open in the standard topology of finite words ($\Omega = \Omega\Sigma^*$).

In the following, let S_f be the $*$-language denoted by $\mathcal{S}$, and as in the previous sections, let S be the ω-language denoted by $\mathcal{S}$ when interpreted as a Buchi automaton.

We recall the approach of [7]. By writing $(\tau)_{S_f}$ for the set of finite words in S_f which are consistent with the observation τ and which end with an observable, the diagnosis function $\mathrm{diag}_\Omega : \Sigma_o^* \to \{\texttt{YES}, \texttt{NO}, \texttt{Don'tKnow}\}$ is defined by

$$\mathrm{diag}_\Omega(\tau) := \begin{cases} \texttt{YES} & \text{if } (\tau)_{S_f} \subseteq L \\ \texttt{NO} & \text{if } (\tau)_{S_f} \cap L = \emptyset \\ \texttt{Don'tKnow} & \text{otherwise.} \end{cases}$$

The comparison between the approach of [7] and the one of Sec. VI-A should be made by correlating diag_Ω and $\mathrm{Diag}_{\Omega\Sigma^\omega}$. Whereas the former focuses only on what happened in the past, the latter one considers information accumulated so far and anticipates on the possible future. An accurate comparison of these two views is out of the

scope of this paper. We rather discuss the relation between the $\approx$-saturation of $\Omega\Sigma^\omega$ in S and the ($\approx$-)*diagnosability* of Ω with respect to S_f [20].

The diagnosability property ensures the "usefulness" of the function diag_Ω; it guarantees that there exists $N \in \mathbb{N}$ such that if the current execution $w \in S_f$ belongs to Ω, then any infinite sequence $P(w) = \tau_0 < \tau_1 < \tau_2 < \dots$ of observations satisfies $\mathrm{diag}_\Omega(\tau_N) = \texttt{YES}$.

Diagnosability therefore fails if one can find arbitrary long pairs of words which are observationally equivalent but which do not agree on membership in Ω. In fact, if infinitely many such pairs exist, by using Koenig's lemma one can exhibit two infinite words w_1 and w_2 in S such that (a) $w_1 \approx w_2$, and (b) no prefix of w_1 reaches Ω whereas almost all prefixes of w_2 do. Notice that the conjunction of (a) and (b) precisely matches the property that $\Omega\Sigma^\omega$ is $\approx$-saturated in S. As a consequence, $\approx$-saturation of $\Omega\Sigma^\omega$ in S and the $\approx$-diagnosability of Ω with respect to S_f are equivalent notions.

As $\Omega\Sigma^\omega$ is an open set, by Lem. 4, $\approx$-openness in S and $\approx$-saturation in S coincide. Th. 13 provides a necessary and sufficient condition to diagnose faulty sequences after finitely many observation steps. However, Th. 13 somehow relaxes the "openness assumption" by requiring only "openness in S". Actually, by Lem. 6, openness and openness in S are almost the same if the system language S is assumed to be closed: it is always possible to replace L with the open set $L \cup S^c$, as $L \cap S = (L \cup S^c) \cap S$. This explains why the so-called "prediagnosability" condition of [9] which corresponds to openness is optimal; if S was not closed, the optimal notion would be "openness in S" instead.

We now turn to the comparison of Algorithm **A1** to decide $\approx$-saturation (in S) and the standard algorithm to decide diagnosability, as originally proposed by [8]. We informally recall this algorithm, according to its generalization in [7]. The central object is a graph called the *twin plant* whose paths denote pairs of $\approx$-equivalent words, and where some vertices are marked. The twin plant construction highly relies on a synchronous product of automata: in this synchronous product, a vertex is *confusing* if in the pair of states it corresponds to, only the first state is final. The twin plant is build in three steps:

Function TP$(\mathcal{S}, \theta)$ // Twin plant construction
Inputs: two finite automata $\mathcal{S}$ and θ (the latter one represents the language Ω and is assumed deterministic)
Outputs: a graph (the twin plant)

1) Build the product automaton $\mathcal{S}_\theta := \mathcal{S} \times \theta$ (a state is final if its second component is final);

2) Abstract away from unobservables in $\mathcal{S}_\theta$ by replacing every sequence of transitions carrying a word of $\Sigma^*\Sigma_o$ by a single transition carrying the unique observable of this word. This yields $\mathrm{OBS}(\mathcal{S}_\theta)$.

3) Return the graph $\mathrm{OBS}(\mathcal{S}_\theta) \times \mathrm{OBS}(\mathcal{S}_\theta)$, where confusing vertices are those whose first component is final whereas the second one is not.

It is not difficult to see that the existence of a reachable cycle in **TP**$(\mathcal{S}, \theta)$ which contains confusing vertices witnesses a counter-example of the $\approx$-diagnosability of Ω with respect to S_f. This leads to the following algorithm.

Algorithm A2 // Diagnosability
Inputs: two finite automata $\mathcal{S}$ and θ (the latter one represents the language Ω and is assumed deterministic)
Outputs: "Ω is not $\approx$-diagnosable with respect to S_f" if the graph **TP**$(\mathcal{S}, \theta)$ contains a cycle of confusing vertices, "Ω is $\approx$-diagnosable with respect to S_f" otherwise.

In fact Algorithm **A2** solves instances of the "$\approx$-saturation in S" problem (solved by **A1**) for which the input language L is open (of the form $\Omega\Sigma^\omega$). The differences between **A1** and **A2** are the following: On the one hand, "$\approx$-saturation in S" is a PSPACE-complete problem (Cor. 12) and Algorithm **A1** is optimal. On the other hand, Algorithm **A2** is quadratic, by searching a cycle in the graph **TP**$(\mathcal{S}, \theta)$ whose size is in $O((|\mathcal{S}||\theta|)^2)$. Although the complexity of **A2** seems considerably lower in that case, the assumption that θ is deterministic is very strong (it hides an exponential time preprocessing procedure to determinize a finite automaton). The twin plant approach hence solves fortunate instances of the saturation problem where rejection of a word by the automaton of Ω is witnessed by a single run. Algorithm **A2** would become "incomplete" for arbitrary non-deterministic automata θ: a cycle of confusing vertices would not characterize a pair of words (w_1, w_2) where $w_1 \approx w_2$, $w_1 \in \Omega\Sigma^\omega$, and $w_2 \notin \Omega\Sigma^\omega$. Because of non-determinism, paths in **TP**$(\mathcal{S}, \theta)$ denote only pairs of runs (ρ_1, ρ_2) over words (w_1, w_2); and in general the fact that ρ_2 is not accepting does not imply $w_2 \notin \Omega\Sigma^\omega$, unless ρ_2 is the unique run. This would be the case if L is a language whose complement is accepted by a deterministic Buchi automaton.

This last remark leads us to propose Algorithm **A3** as an extension of Algorithm **A2** from the class of open languages to the class of languages whose complement is deterministic Buchi definable. This class is characterized in the Borel hierarchy as Σ_2 which contains sets obtained by a countable union of closed sets. Membership in Σ_2 is decidable [15, Chapter I, Proposition 7.10]. For this strictly larger class of languages, confusing cycles become cycles of the form $\{(q_1, q'_1), (q_2, q'_2), \dots, (q_k, q'_k)\}$ where q_i is accepting and all the q'_j's are rejecting. Because open sets are strictly contained in Σ_2 [15, Chapter III, Proposition 2.9], Algorithm **A3** is a true extension of Algorithm **A2**, but is still quadratic, although it solves instances of the "$\sim$-saturation" problem (where $\sim$ is an observational equivalence and $L \in \Sigma_2$).

VII. Conclusion

We have introduced the problems of saturation and openness on the space of infinite words, shown their decidability, and studied their complexity; the former is PSPACE-complete and the latter one is EXPTIME. Recall that no lower bound for the openness problem is know when the

input is given as a non-deterministic Buchi automaton. Also, we are not aware of any decision procedure the "openness in S" problem in general, but by assuming that S is a closed set, this problem reduces to the one of openness.

Finally, we have investigated diagnosis of ω-regular properties and its relation with classic diagnosis. More specifically, we have shown that the standard algorithm for checking diagnosability is a particular case of the saturation problem corresponding to properties that are open sets.

For future work, we believe the rational ω-relations $\approx$ and R (Th. 11) are central objects for the main following reasons.

- The relation $\approx^c$, the complement of $\approx$, is also rational whereas rational relations are not closed under complement in general.
- As finiteness of rational relations is decidable, the number of confusing cycles can be estimated.
- Although a naive extension of $\approx$-saturation to a decentralized setting would fail (as rational relations are not closed under intersection), we can tune the devices to decide the co-observability property [19], [26]. Also, by scanning the relation R, we can exhibit strategic situations where observation capabilities can be augmented to achieve an objective (e.g. by a communication mechanism), in the spirit of [18].

Advisedly exploiting these aspects is ongoing work.

REFERENCES

[1] B. Alpern and F. B. Schneider. Recognizing safety and liveness. *Distributed Computing*, 2:117–126, 1987.

[2] Jean Berstel. *Transductions and Context-Free Languages*. Teubner Studienbücher, Stuttgart, 1979.

[3] C. C. Elgot and J. E. Mezei. On relations defined by generalized finite automata. *IBM Journal of Research and Development*, 9(January):47–68, 1965.

[4] Christiane Frougny and Jacques Sakarovitch. Synchronized rational relations of finite and infinite words. *Theor. Comput. Sci.*, 108(1):45–82, 1993.

[5] Françoise Gire and Maurice Nivat. Relations rationnelles infinitaires. *Calcolo*, 21(2):91–125, 1984.

[6] E. Grädel, W. Thomas, and T. Wilke, editors. *Automata, Logics, and Infinite Games: A Guide to Current Research [outcome of a Dagstuhl seminar, February 2001]*, volume 2500 of *Lecture Notes in Computer Science*. Springer, 2002.

[7] T. Jeron, H. Marchand, S. Pinchinat, and M-O. Cordier. Supervision patterns in discrete event systems diagnosis. In *8th Workshop on Discrete Event Systems, WODES'06*, Ann Arbor, Michigan, USA, July 2006.

[8] S. Jiang, Z. Huang, V. Chandra, and R. Kumar. A polynomial time algorithm for diagnosability of discrete event systems. *IEEE Transactions on Automatic Control*, 46(8):1318–1321, 2001.

[9] S. Jiang and R. Kumar. Failure diagnosis of discrete event systems with linear-time temporal logic fault specifications. *IEEE Transactions on Automatic Control*, 49(6):934–945, 2004.

[10] Nils Klarlund. Progress measures for complementation of omega-automata with applications to temporal logic. In *FOCS*, pages 358–367. IEEE, 1991.

[11] Orna Kupferman and Moshe Y. Vardi. Model checking of safety properties. *Formal Methods in System Design*, 19(3):291–314, 2001.

[12] Lawrence H. Landweber. Decision problems for omega-automata. *Mathematical Systems Theory*, 3(4):376–384, 1969.

[13] R. McNaughton. Testing and generating infinite sequences by a finite automaton. *Information and Control*, 9:521–530, 1966.

[14] Doron Peled, Thomas Wilke, and Pierre Wolper. An algorithmic approach for checking closure properties of ω-regular languages. *Theoretical Computer Science*, 195(2):183–203, 1998.

[15] Dominique Perrin and Jean-Eric Pin. *Infinite words, automata, semigroups, logic and games*. Elsevier, 2004.

[16] C. Prieur. *Fonctions rationnelles de mots infinis et continuté*. Thèse de Doctorat, Univ. Paris 7, June 2000.

[17] M. O. Rabin and D. Scott. Finite automata and their decision problems. *IBM Journal of Research and Development*, 3:114–125, 1959.

[18] L. Ricker and B. Caillaud. Mind the gap: Expanding communication options in decentralized discrete-event control. In *46th IEEE Conference on Decision and Control*, New Orleans, LA, USA, December 2007.

[19] K. Rudie and W. M. Wonham. Think globally, act locally: Decentralized supervisory control. *IEEE Trans. Autom. Control*, 37(11):1692–1708, November 1992.

[20] M. Sampath, R. Sengupta, S. Lafortune, K. Sinaamohideen, and D. Teneketzis. Diagnosability of discrete event systems. *IEEE Transactions on Automatic Control*, 40(9):1555–1575, 1995.

[21] M. Sampath, R. Sengupta, S. Lafortune, K. Sinaamohideen, and D. Teneketzis. Failure diagnosis using discrete event models. *IEEE Transactions on Control Systems Technology*, 4(2):105–124, March 1996.

[22] Walter J. Savitch. Relationships between nondeterministic and deterministic tape complexities. *J. Comput. System. Sci.*, 4:177–192, 1970.

[23] A. P. Sistla, M. Vardi, and P. Wolper. The complementation problem for Buchï automata with applications to temporal logic. *Theoretical Computer Science*, 49:217–237, 1987.

[24] A. Prasad Sistla. Safety, liveness and fairness in temporal logic. *Formal Asp. Comput.*, 6(5):495–512, 1994.

[25] L. Staiger. ω-languages. In Grzegorz Rozenberg and Arto Salomaa, editors, *Handbook of formal languages, vol. 3: beyond words*, chapter 10, pages 339–388. Springer-Verlag New York, Inc., New York, NY, USA, 1997.

[26] Tae-Sic Yoo and Stephane Lafortune. A general architecture for decentralized supervisory control of discrete-event systems. *Discrete event dynamic systems : theory and applications*, July 2002.